Automated Constructivization of Proofs

Frédéric Gilbert

Automated Constructivization of Proofs

Frédéric Gilbert

École des Ponts ParisTech, Inria, CEA LIST
frederic.a.gilbert@inria.fr

Abstract. No computable function can output a constructive proof from a classical one whenever its associated theorem also holds constructively. We show in this paper that it is however possible, in practice, to turn a large amount of classical proofs into constructive ones. We describe for this purpose a linear-time constructivization algorithm which is provably complete on large fragments of predicate logic.

1 Introduction

Classical and constructive provability match on several specific sets of propositions. In propositional logic, as a consequence of Glivenko's theorem [1], a formula $\neg A$ is a classical theorem iff it is a constructive one. In arithmetic, a Π^0_2 proposition is a theorem in Peano arithmetic iff it is a theorem in Heyting arithmetic [2].

We present in this paper an efficient constructivization algorithm Construct for predicate logic in general, from cut-free classical sequent calculus **LK** to constructive sequent calculus **LJ**. Unlike the two previous examples, constructivization in predicate logic is as hard as constructive theorem proving. Therefore, as we expect Construct to terminate, Construct is incomplete in the sense that it may terminate with a failure output.

Construct consists of three **linear-time** steps:

1. An algorithm Normalize, designed to push occurrences of the right weakening rule towards the root in **LK** proofs. Its purpose is to limit the number of propositions appearing at the right-hand side of sequents in **LK** proofs.
2. A partial translation from cut-free **LK** to a new constructive system **LI**. This algorithm is referred to as Annotate as the **LI** system is designed as **LK** equipped with specific annotations – making it a constructive system. Annotate is the only step which may fail.
3. A complete translation Interpret from **LI** to **LJ**.

The Normalize step taken alone leads to a simple yet efficient constructivization algorithm Weak construct, which is defined to succeed whenever the result of Normalize happens to be directly interpretable in **LJ**, i.e. to have at most one proposition on the right-hand side of sequents in its proof.

The main property of Construct is to be provably **complete** on large fragments of predicate logic, in the sense that for any proposition A in one of these fragments, CONSTRUCT is ensured to terminate successfully on any cut-free **LK** proof of A. Such fragments for which classical and constructive provability match will be referred to as **constructive fragments**. For instance, as a consequence of Glivenko's theorem [1], the set of negated propositions is a **constructive fragment** of propositional logic. The completeness properties of CONSTRUCT lead to the following results:

– The identification of a new constructive fragment F, the fragment of assertions containing no negative occurrence of the connective $\vee$ and no positive occurrence of the connective $\Rightarrow$. Both WEAK CONSTRUCT and CONSTRUCT are provably complete on F.

– The completeness of CONSTRUCT on two already known constructive fragments. The first one, referred to as F_{Ku}, appears as the set of fix points of a polarized version of Kuroda's double-negation translation [3, 4]. The second one, referred to as F_{Ma}, appears as a set of assertions for which any cut-free **LK** proof can be directly interpreted as a proof in Maehara's multi-succedent calculus [5]. Hence, the completeness of CONSTRUCT on these two fragments yields a uniform proof of two results coming from very different works.

After the introduction of basic notations and definitions, the two already known constructive fragments F_{Ku} and F_{Ma} are presented. Then, the NORMALIZE step is presented along with the simple constructivization algorithm WEAK CONSTRUCT. In the following section, the new constructive fragment F is defined, and WEAK CONSTRUCT is proved complete on F. Then, the full constructivization algorithm CONSTRUCT is introduced together with the proof of its completeness on F, F_{Ku} and F_{Ma}. In the last part, experimental results of constructivization using WEAK CONSTRUCT and CONSTRUCT are presented. These experiments are based the classical theorem prover **Zenon** [10] and the constructive proof checker **Dedukti** [9].

2 Notations and definitions

In the following, we only consider as primitive the connectives and quantifiers $\forall$, $\exists$, $\wedge$, $\vee$, $\Rightarrow$ and $\bot$. $\neg A$ is defined as $A \Rightarrow \bot$. $\top$, which doesn't appear in this paper, could be defined as $\bot \Rightarrow \bot$.

We use a definition of sequents based on **multisets**. The size of a multiset Γ will be referred to as $|\Gamma|$. We will use the notation (A) to refer to a multiset containing either zero or one element. Given a multiset $\Gamma = A_1, \cdots, A_n$, we will use the notations $\neg\Gamma$ and $\Gamma \Rightarrow B$ as shorthands for $\neg A_1, \cdots, \neg A_n$, and $A_1 \Rightarrow B, \cdots, A_n \Rightarrow B$ respectively. Finally, we use the notation $\bigvee$ to refer to an arbitrary encoding of the n-ary disjunction from the binary one – using $\bot$ for the nullary case.

Definition 1. *We define the cut-free classical sequent calculus **LK** with the following rules:*

$$\frac{}{\bot \vdash} \bot_L \quad \frac{}{A \vdash A} \textit{axiom}$$

$$\frac{\Gamma \vdash \Delta}{\Gamma, \Gamma' \vdash \Delta} \textit{weak}_L \quad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, \Delta'} \textit{weak}_R$$

$$\frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta} \textit{contr}_L \quad \frac{\Gamma \vdash A, A, \Delta}{\Gamma \vdash A, \Delta} \textit{contr}_R$$

$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta} \wedge_L \quad \frac{\Gamma \vdash A, \Delta \qquad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta} \wedge_R$$

$$\frac{\Gamma, A \vdash \Delta \qquad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta} \vee_L \quad \frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \vee B, \Delta} \vee_R$$

$$\frac{\Gamma \vdash A, \Delta \qquad \Gamma, B \vdash \Delta}{\Gamma, A \Rightarrow B \vdash \Delta} \Rightarrow_L \quad \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \Rightarrow B, \Delta} \Rightarrow_R$$

$$\frac{\Gamma, A[t/x] \vdash \Delta}{\Gamma, \forall x A \vdash \Delta} \forall_L \quad \frac{\Gamma \vdash A, \Delta}{\Gamma \vdash \forall x A, \Delta} \forall_R$$

$$\frac{\Gamma, A \vdash \Delta}{\Gamma, \exists x A \vdash \Delta} \exists_L \quad \frac{\Gamma \vdash A[t/x], \Delta}{\Gamma \vdash \exists x A, \Delta} \exists_R$$

with the standard freshness constraints for the variables introduced in the rules $\forall_R$ and $\exists_L$.

Definition 2. *We define the constructive sequent calculus **LJ** from **LK**, applying the following changes:*

- *All rules except $\textit{contr}_R$, $\vee_R$, $\Rightarrow_L$ are restricted to sequents with at most one proposition on the right-hand side of sequents.*

  For instance, $\wedge_R$ becomes $\dfrac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \wedge_R$
- *There is no $\textit{contr}_R$ rule*
- *The $\vee_R$ rule is split into two rules* $\dfrac{\Gamma \vdash A_i}{\Gamma \vdash A_0 \vee A_1} \vee_R$
- *The $\Rightarrow_L$ rule becomes* $\dfrac{\Gamma \vdash A \qquad \Gamma, B \vdash (C)}{\Gamma, A \Rightarrow B \vdash (C)} \Rightarrow_L$

– *We add a cut rule* $\dfrac{\Gamma \vdash A \qquad \Gamma, A \vdash (B)}{\Gamma \vdash (B)}\,cut$

Remark 1. In these presentations of **LK** and **LJ**,

– weakenings are applied to multisets instead of propositions

– $\perp_L$ and *axiom* are not relaxed to $\dfrac{}{\Gamma, \perp \vdash \Delta}\,\perp_L$ and $\dfrac{}{\Gamma, A \vdash A, \Delta}\,axiom$

These specific conventions are chosen to ease the definition of the algorithm Normalize in Section 5, which requires pushing weakenings towards the root of the proof.

Definition 3. *We introduce the following notations in **LK**, along with their constructive analogs in **LJ**:*

– $\dfrac{}{\Gamma, A \vdash A, \Delta}\,axiom^*$ *for* $\dfrac{\dfrac{\dfrac{}{A \vdash A}\,axiom}{\Gamma, A \vdash A}\,weak_L}{\Gamma, A \vdash A, \Delta}\,weak_R$

– $\dfrac{}{\Gamma, \perp \vdash \Delta}\,\perp_L^*$ *for* $\dfrac{\dfrac{\dfrac{}{\perp \vdash}\,\perp_L}{\Gamma, \perp \vdash}\,weak_L}{\Gamma, \perp \vdash \Delta}\,weak_R$

– $\dfrac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta}\,\neg_L$ *for* $\dfrac{\Gamma \vdash A, \Delta \qquad \dfrac{}{\Gamma, \perp \vdash \Delta}\,\perp_L^*}{\Gamma, \neg A \vdash \Delta}\,\Rightarrow_L$

– $\dfrac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta}\,\neg_R$ *for* $\dfrac{\dfrac{\Gamma, A \vdash \Delta}{\Gamma, A \vdash \perp, \Delta}\,weak_R}{\Gamma \vdash \neg A, \Delta}\,\Rightarrow_R$

3 State of the art: two constructive fragments of predicate logic

Constructive sequent calculus – as well as constructive natural deduction – extends the notion of constructive provability from propositions to sequents of the shape $\Gamma \vdash (G)$, which will be referred to as **mono-succedent sequents**. As a consequence, we will define constructive fragments of predicate logic as sets of mono-succedent sequents instead of sets of simple propositions.

The definitions of these fragments will be based on the usual notion of polarity of occurrences of connectives, quantifiers and atoms in a sequent: given a sequent $\Gamma \vdash \Delta$,

- the root of a proposition in Γ is negative, the root of a proposition in Δ is positive
- polarity only changes between an occurrence of $A \Rightarrow B$ and the occurrence of its direct subformula A (in particular, as $\neg A$ is defined as $A \Rightarrow \bot$, it changes between $\neg A$ and its direct subformula A)

Definition 4. *We define the following fragments of predicate logic:*

- F_{Ku}*, the fragment of sequents of the shape* $\Gamma \vdash$ *containing no positive occurrence of* $\forall$*.*
- F_{Ma}*, the fragment of mono-succedent sequents containing no positive occurrence of* $\forall$ *and no positive occurrence of* $\Rightarrow$*.*

Theorem 1. F_{Ku} *is a constructive fragment of predicate logic: for any sequent* $\Gamma \vdash$ *in* F_{Ku}*,* $\Gamma \vdash$ *is classically provable iff it is constructively provable.*

The key arguments to prove this theorem as an adaptation of Kuroda's double negation translation [3] are the following:

1. Kuroda's double negation translation [3] is based on a double negation translation $|\cdot|_{Ku}$ inserting double-negations after any occurrence of $\forall$. The original theorem is that a proposition A is classically provable iff $\neg\neg|A|_{Ku}$ is constructively provable.
2. It can adapted in two ways. First, $|\cdot|_{Ku}$ can be lightened to insert double negations only after positive occurrences of $\forall$ as shown in [4], and extended from propositions to contexts. Second, the main statement can be turned to the following one: a classical sequent $\Gamma \vdash \Delta$ is classically provable iff $|\Gamma, \neg\Delta|_{Ku} \vdash$ is constructively provable
3. By definition of F_{Ku}, a sequent $\Gamma \vdash$ in F_{Ku} admits the property $\Gamma = |\Gamma|_{Ku}$, hence $\Gamma \vdash$ is classically provable iff it is constructively provable.

We don't give more details on this proof as the completeness of CONSTRUCT on F_{Ku} shown in Section 6 will yield a new proof of this result.

Remark 2. One could expect similar constructive fragments to be found using other double negation translations, such as Gödel-Gentzen's [7, 6] or Kolmogorov's [8]. Unfortunately, these two translations always insert double-negations in front of atoms, hence they cannot be easily modified to leave a large fragment of propositions unchanged.

Theorem 2. F_{Ma} *is a constructive fragment of predicate logic: for any sequent* $\Gamma \vdash (G)$ *in* F_{Ma}*,* $\Gamma \vdash (G)$ *is classically provable iff it is constructively provable.*

It lays on a key idea: polarity restrictions have a direct influence on the shape of cut-free proofs. It can be presented in the following way:

Lemma 1. *For any connective or quantifier X and any cut-free **LK** proof Π of a sequent $\Gamma \vdash \Delta$:*

- *If $\Gamma \vdash \Delta$ contains no positive occurrence of X, then Π doesn't contain the rule X_R.*
- *If $\Gamma \vdash \Delta$ contains no negative occurrence of X, then Π doesn't contain the rule X_L.*

This lemma can be proved directly by induction on cut-free **LK** proofs. Using this lemma, the key arguments to prove Theorem 2 are the following:

1. All **LK** rules except $\Rightarrow_R$ and $\forall_R$ rules belong in Maehara's multi-succedent calculus [5], a constructive multi-succedent sequent calculus.
2. By lemma 1, F_{Ma} sequents are proved by cut-free **LK** proofs without the $\Rightarrow_R$ and $\forall_R$ rules.
3. Hence, a sequent $\Gamma \vdash (G)$ in F_{Ma} is classically provable iff it is constructively provable.

Again, we don't give more details on this proof as the completeness of Construct on F_{Ma} shown in Section 6 will yield a new proof of this result.

Remark 3. The same fragment F_{Ma} can be found using similar multi-succedent constructive systems, such as Dragalin's calculus GHPC [11].

4 The weakening normalization

A naive constructivization algorithm can be defined by selecting **LK** proofs which can be directly interpreted in **LJ**.

In this direct interpretation, premises of the classical rules $\vee_R$ and $\Rightarrow_L$ may be multi-succedent only when they are introduced by a $weak_R$ whose premise is a mono-succedent sequent. For instance, the classical derivation

$$\dfrac{\dfrac{\Gamma \vdash A}{\Gamma \vdash A, B}\, weak_R}{\Gamma \vdash A \vee B}\, \vee_R \quad \text{can be interpreted as} \quad \dfrac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\, \vee_R \ .$$

However, in practice, the $weak_R$ rule doesn't appear as low as possible – in presentations using multi-succedents *axiom* rules, they may not appear at all. Such situations are problematic for constructive interpretations: for instance, a classical proof such as

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A}\, axiom}{A \vdash A, B}\, weak_R}{\vdash A \Rightarrow A, B}\, \Rightarrow_R}{\vdash (A \Rightarrow A) \vee B}\, \vee_R$$

cannot be interpreted in **LJ** directly because the $weak_R$ rule doesn't occur immediately above the $\vee_R$ rule.

The NORMALIZE algorithm is designed to address this issue, pushing the application of $weak_R$ as low as possible in proofs. In its definition, we need to consider all possible configuration of $weak_R$ appearing above a **LK** rule. In order to factor this definition, we partition all such configurations into three classes **A**, **B**, and **C**.

These definitions will be based on the following notation of **LK** proofs:

Definition 5. *We write any cut-free **LK** rule X as*

$$\frac{\Gamma, L_1 \vdash R_1, \Delta \qquad \cdots \qquad \Gamma, L_n \vdash R_n, \Delta}{\Gamma, L \vdash R, \Delta}\ X$$

*where $L_1, \cdots, L_n, R_1, \cdots, R_n, L$ and R are the (possibly empty) **multisets** of propositions containing the active propositions of the rule X.*

For instance, in the rule $\dfrac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \Rightarrow B, \Delta} \Rightarrow_R$,

$L_1 = \{A\}$, $R_1 = \{B\}$, $L = \emptyset$, and $R = \{A \Rightarrow B\}$.

The classes **A**, **B**, and **C** are defined as follows:

Definition 6. *We consider all configurations where $weak_R$ appears above a **LK** rule X, in its i-th premise:*

$$\frac{\cdots \qquad \dfrac{\Gamma, L_i \vdash \Delta_i}{\Gamma, L_i \vdash R_i, \Delta}\ weak_R \qquad \cdots}{\Gamma, L \vdash R, \Delta}\ X$$

This weakening can be done on propositions in R_i, in Δ or both: in the general case, we only know $\Delta_i \subseteq (R_i, \Delta)$. We define the following partition of all cases:

- ***A**: $R_i \subseteq \Delta_i$*
- ***B**: $R_i \not\subseteq \Delta_i$ and $\Delta_i \subseteq \Delta$*
- ***C**: $R_i \not\subseteq \Delta_i$ and $\Delta_i \not\subseteq \Delta$. This only happens when $|R_i| = 2$, when exactly one proposition of R_i is in Δ_i.*

Definition 7. NORMALIZE *is a linear-time algorithm associating any cut-free **LK** proof of a sequent $\Gamma \vdash \Delta$ to a proof of a sequent $\Gamma \vdash \Delta'$, where $\Delta' \subseteq \Delta$. It is defined recursively. Using the conventions of Definition 5, we describe the original proof Π as*

$$\frac{\dfrac{\Pi_1}{\Gamma, L_1 \vdash R_1, \Delta} \qquad \cdots \qquad \dfrac{\Pi_n}{\Gamma, L_n \vdash R_n, \Delta}}{\Gamma, L \vdash R, \Delta}\ X$$

The definition of NORMALIZE*(Π) is based on the analysis of the proof*

$$\dfrac{\dfrac{\dfrac{\text{NORMALIZE}(\Pi_1)}{\Gamma, L_1 \vdash \Delta_1}}{\Gamma, L_1 \vdash R_1, \Delta}\,weak_R \quad \dots \quad \dfrac{\dfrac{\text{NORMALIZE}(\Pi_n)}{\Gamma, L_n \vdash \Delta_n}}{\Gamma, L_n \vdash R_n, \Delta}\,weak_R}{\Gamma, L \vdash R, \Delta}\,X$$

The different cases are the following:

- *Case **1**: for all index i, **A** holds, i.e. $R_i \subseteq \Delta_i$.*
  If X is $weak_R$, we define NORMALIZE*(Π) as* NORMALIZE*(Π_1).*

  Else, writing $\Delta_i = R_i, \Delta'_i$, we define NORMALIZE*($\Pi$) as*

  $$\dfrac{\dfrac{\dfrac{\text{NORMALIZE}(\Pi_1)}{\Gamma, L_1 \vdash R_1, \Delta'_1}}{\Gamma, L_1 \vdash R_1, \Delta'}\,weak_R \quad \dots \quad \dfrac{\dfrac{\text{NORMALIZE}(\Pi_n)}{\Gamma, L_n \vdash R_n, \Delta'_n}}{\Gamma, L_n \vdash R_n, \Delta'}\,weak_R}{\Gamma, L \vdash R, \Delta'}\,X$$

  where Δ' is the smallest multiset containing all multisets Δ'_i

- *Case **2**: there exists a smallest premise i for which **B** holds, i.e. $R_i \not\subseteq \Delta_i$ and $\Delta_i \subseteq \Delta$. As $R_i \neq \emptyset$, ether X is $\Rightarrow_R$ or $L_i = \emptyset$.*

  If X is $\Rightarrow_R$, we define NORMALIZE*(Π) as* $\dfrac{\dfrac{\dfrac{\text{NORMALIZE}(\Pi_1)}{\Gamma, A \vdash \Delta_1}}{\Gamma, A \vdash B, \Delta_1}\,weak_R}{\Gamma \vdash A \Rightarrow B, \Delta_1}\,\Rightarrow_R$

  Else, $L_i = \emptyset$ and we define NORMALIZE*(Π) as* $\dfrac{\dfrac{\text{NORMALIZE}(\Pi_i)}{\Gamma \vdash \Delta_i}}{\Gamma, L \vdash \Delta_i}\,weak_L$

- *Case **3**: there exists a smallest premise i for which the case **C** applies, i.e. $R_i \not\subseteq \Delta_i$ and $\Delta_i \not\subseteq \Delta$. This only happens when $|R_i| = 2$, when exactly one proposition of R_i is in Δ_i. In this case, X is either $contr_R$ or $\vee_R$.*

  If X is $contr_R$, we can write $R_1 = A, A$, and $\Delta_1 = (A, \Delta'_1)$ with $\Delta'_1 \subseteq \Delta$. We define NORMALIZE*($\Pi$) as* NORMALIZE*($\Pi_1$).*

  If X is $\vee_R$, we can write $R_1 = A_0, A_1$, and $\Delta_1 = (A_k, \Delta'_1)$ with $\Delta'_1 \subseteq \Delta$.

  We define NORMALIZE*(Π) as* $\dfrac{\dfrac{\dfrac{\text{NORMALIZE}(\Pi_1)}{\Gamma \vdash A_k, \Delta'_1}}{\Gamma \vdash A_0, A_1, \Delta'_1}\,weak_R}{\Gamma \vdash A_0 \vee A_1, \Delta'_1}\,\vee_R$

Remark 4. The nullary rules *axiom* and $\bot_L$ having no premise, they match the first case.

Definition 8. *We define a first constructivization algorithm* Weak construct*, which*

- *takes as input a cut-free* ***LK*** *proof* $\dfrac{\Pi}{\Gamma \vdash (G)}$ *,*
- *computes the proof* $\dfrac{\textsc{Normalize}(\Pi)}{\Gamma \vdash (G)}\,weak_R$ *,*
- *outputs its* ***LJ*** *interpretation if it exists and fails otherwise*

5 A new constructive fragment

Definition 9. *We define F as the fragment of mono-succedent sequents containing no negative occurrence of $\vee$ and no positive occurrence of $\Rightarrow$.*

Theorem 3. Weak construct *is complete on F: if Π is a cut-free* ***LK*** *proof of a sequent $\Gamma \vdash (G) \in F$, then* Weak construct*(Π) succeeds.*

Proof. By Lemma 1, F sequents are proved by cut-free **LK** proofs containing no $\vee_L$ or $\Rightarrow_R$ rule. We prove that for any such proof Π, Normalize(Π) proves a mono-succedent sequent interpretable in **LJ**. This proof is done by induction on cut-free **LK** proofs containing no $\vee_L$ or $\Rightarrow_R$ rule, following the partition of cases and the notations introduced in the definition of Normalize:

- Case **1**: we split this case according to the rule X.
  - nullary rules: *axiom* and $\bot_L$ are interpretable in **LJ**.
  - $weak_R$: The result follows directly by induction hypothesis.
  - other unary rules: In these cases $\Delta' = \Delta'_1$, hence Normalize(Π) is

$$\dfrac{\dfrac{\dfrac{\textsc{Normalize}(\Pi_1)}{\Gamma, L_1 \vdash R_1, \Delta'_1}}{\Gamma, L_1 \vdash R_1, \Delta'_1}\,weak_R}{\Gamma, L \vdash R, \Delta'_1}\,X$$

    By induction hypothesis, Normalize(Π_1) is interpretable in **LJ**. Hence, $|R_1| \leq 1$, which ensures that X is neither $contr_R$ nor $\vee_R$. All other unary rules lead to a proof interpretable in **LJ**, therefore the result is interpretable in **LJ**.

  - $\vee_L$: This case doesn't occur by hypothesis

- $\Rightarrow_L$: By induction hypothesis, NORMALIZE(Π_1) and NORMALIZE(Π_2) are interpretable in **LJ**, hence $|R_1, \Delta'_1| \leq 1$. As $|R_1| = 1$, $\Delta'_1 = \emptyset$, and $\Delta' = \Delta'_2$.

  As $\dfrac{\dfrac{\Gamma \vdash A}{\Gamma \vdash A, \Delta'_2}\, weak_R \qquad \dfrac{\Gamma, B \vdash \Delta'_2}{\Gamma, B \vdash \Delta'_2}\, weak_R}{\Gamma, A \Rightarrow B \vdash \Delta'_2} \Rightarrow_L$

  is interpretable as $\dfrac{\Gamma \vdash A \qquad \Gamma, B \vdash \Delta'_2}{\Gamma, A \Rightarrow B \vdash \Delta'_2} \Rightarrow_L$ in **LJ**, the result follows.

- $\wedge_R$: By induction hypothesis, NORMALIZE(Π_1) and NORMALIZE(Π_2) are interpretable in **LJ**, hence $|R_1, \Delta'_1| \leq 1$ and $|R_2, \Delta'_2| \leq 1$. As $|R_1| = |R_2| = 1$, $\Delta'_1 = \Delta'_2 = \emptyset$. Therefore $\Delta' = \emptyset$, from which the result follows.

– Case **2**: By hypothesis, X is not $\Rightarrow_R$, hence NORMALIZE(Π) is defined as

$$\dfrac{\dfrac{\text{NORMALIZE}(\Pi_i)}{\Gamma \vdash \Delta_i}}{\Gamma, L \vdash \Delta_i}\, weak_L$$

The result follows by induction hypothesis.

– Case **3**: If X is $contr_R$, the result follows directly by induction hypothesis. Else, X is $\vee_R$. By induction hypothesis, NORMALIZE(Π_1) is interpretable in **LJ**, thus $|A_k, \Delta'_1| \leq 1$, and $\Delta'_1 = \emptyset$.

As $\dfrac{\dfrac{\Gamma \vdash A_k}{\Gamma \vdash A_0, A_1}\, weak_R}{\Gamma \vdash A_0 \vee A_1} \vee_R$ is interpretable as $\dfrac{\Gamma \vdash A_k}{\Gamma \vdash A_0 \vee A_1} \vee_R$ in **LJ**,

the result follows.

Corollary 1. *The fragment F is a constructive fragment of predicate logic: a sequent $\Gamma \vdash (G)$ is classically provable iff it is constructively provable.*

6 The full constructivization algorithm

The previous algorithm WEAK CONSTRUCT was based on the reject of multi-succedent sequents. The idea leading to our main algorithm CONSTRUCT is to try to interpret multi-succedent sequents constructively as well. This interpretation is based on a new multi-succedent constructive system, which will be referred to as **LI** in the following. As mentioned in the introduction, the constructivization algorithm CONSTRUCT comprises three steps: first the algorithm NORMALIZE, then a partial translation ANNOTATE from **LK** proofs to **LI** proofs, and finally a complete translation INTERPRET from **LI** proof to **LJ** proofs.

There are several ways to interpret multi-succedent sequents constructively. For instance, $\Gamma \vdash \bigvee \Delta$ and $\Gamma, \neg\Delta \vdash$ are two possible interpretations of a multi-succedent sequent $\Gamma \vdash \Delta$. These interpretation are equivalent classically but not constructively: for instance, the classical sequent $\vdash A, \neg A$ is not provable constructively under the first interpretation, but it is provable constructively under the second one. As a consequence, some classical rules may be constructively valid or not according to the chosen interpretation of classical sequents.

The new system **LI** is built to benefit from the freedom left in the constructive interpretation of classical sequents. **LI** is designed as a sequent calculus based on **annotated sequents**, where the annotation will refer to the choice of constructive interpretation of the underlying classical sequent. We formalize first the notion of **annotated sequents**.

Definition 10. *We define the set of* ***annotated sequents*** *as sequents of the shape* $\Gamma \vdash \Delta_1; \Delta_2$.

We define the following interpretation INTERPRET *on annotated sequents:*
$\textsc{Interpret}(\Gamma \vdash \Delta_1; \Delta_2) \quad = \quad \Gamma, \neg\Delta_2 \vdash \bigvee \Delta_1.$
In the following, this function will be extended from ***LI*** *proofs to* ***LJ*** *proofs.*

We define the following erasure function ERASE *on annotated sequents:*
$\textsc{Erase}(\Gamma \vdash \Delta_1; \Delta_2) \quad = \quad \Gamma \vdash \Delta_1, \Delta_2.$

In the following, this function will be extended from ***LI*** *proofs to* ***LK*** *proofs.*

Then, we define the system **LI** in the following way:

Definition 11. ***LI*** *is based on the following rules:*

$$\frac{}{\bot \vdash ;}\ \bot_L \qquad \frac{}{A \vdash A;}\ axiom^1 \qquad \frac{}{A \vdash ; A}\ axiom^2$$

$$\frac{\Gamma \vdash \Delta_1; \Delta_2}{\Gamma, \Gamma' \vdash \Delta_1; \Delta_2}\ weak_L \qquad \frac{\Gamma \vdash \Delta_1; \Delta_2}{\Gamma \vdash \Delta_1, \Delta_1'; \Delta_2, \Delta_2'}\ weak_R$$

$$\frac{\Gamma, A, A \vdash \Delta_1; \Delta_2}{\Gamma, A \vdash \Delta_1; \Delta_2}\ contr_L \qquad \frac{\Gamma \vdash A, A, \Delta_1; \Delta_2}{\Gamma \vdash A, \Delta_1; \Delta_2}\ contr_R^1 \qquad \frac{\Gamma \vdash \Delta_1; A, A, \Delta_2}{\Gamma \vdash \Delta_1; A, \Delta_2}\ contr_R^2$$

$$\frac{\Gamma, A, B \vdash \Delta_1; \Delta_2}{\Gamma, A \wedge B \vdash \Delta_1; \Delta_2}\ \wedge_L \qquad \frac{\Gamma \vdash A, \Delta_1; \Delta_2 \qquad \Gamma \vdash B, \Delta_1; \Delta_2}{\Gamma \vdash A \wedge B, \Delta_1; \Delta_2}\ \wedge_R^1$$

$$\frac{\Gamma \vdash ; A, \Delta_2 \qquad \Gamma \vdash ; B, \Delta_2}{\Gamma \vdash ; A \wedge B, \Delta_2}\ \wedge_R^2 \qquad \frac{\Gamma \vdash A, \Delta_1; \Delta_2 \qquad \Gamma \vdash B, \Delta_1; \Delta_2}{\Gamma \vdash \Delta_1; A \wedge B, \Delta_2}\ \wedge_R^3,\ |\Delta_1| \geq 1$$

$$\frac{\Gamma, A \vdash \Delta_1; \Delta_2 \qquad \Gamma, B \vdash \Delta_1; \Delta_2}{\Gamma, A \vee B \vdash \Delta_1; \Delta_2}\ \vee_L$$

$$\frac{\Gamma \vdash A, B, \Delta_1; \Delta_2}{\Gamma \vdash A \vee B, \Delta_1; \Delta_2}\ \vee_R^1 \qquad \frac{\Gamma \vdash \Delta_1; A, B, \Delta_2}{\Gamma \vdash \Delta_1; A \vee B, \Delta_2}\ \vee_R^2$$

$$\frac{\Gamma \vdash ; A, \Delta_2 \qquad \Gamma, B \vdash ; \Delta_2}{\Gamma, A \Rightarrow B \vdash ; \Delta_2}\ \Rightarrow_L^1 \qquad \frac{\Gamma \vdash A, \Delta_1; \Delta_2 \qquad \Gamma, B \vdash \Delta_1; \Delta_2}{\Gamma, A \Rightarrow B \vdash \Delta_1; \Delta_2}\ \Rightarrow_L^2,\ |\Delta_1| \geq 1$$

$$\frac{\Gamma, A \vdash B; \Delta_2}{\Gamma \vdash A \Rightarrow B; \Delta_2} \Rightarrow_R^1 \quad \frac{\Gamma, A \vdash ; B, \Delta_2}{\Gamma \vdash ; A \Rightarrow B, \Delta_2} \Rightarrow_R^2$$

$$\frac{\Gamma, A[t/x] \vdash \Delta_1; \Delta_2}{\Gamma, \forall x A \vdash \Delta_1; \Delta_2} \forall_L \quad \frac{\Gamma \vdash A; \Delta_2}{\Gamma \vdash \forall x A; \Delta_2} \forall_R^1 \quad \frac{\Gamma \vdash A; \Delta_2}{\Gamma \vdash ; \forall x A, \Delta_2} \forall_R^2$$

$$\frac{\Gamma, A \vdash \Delta_1; \Delta_2}{\Gamma, \exists x A \vdash \Delta_1; \Delta_2} \exists_L \quad \frac{\Gamma \vdash A[t/x], \Delta_1; \Delta_2}{\Gamma \vdash \exists x A, \Delta_1; \Delta_2} \exists_R^1 \quad \frac{\Gamma \vdash \Delta_1; A[t/x], \Delta_2}{\Gamma \vdash \Delta_1; \exists x A, \Delta_2} \exists_R^2$$

with the standard freshness constraints for the variables introduced in the rules $\forall_R^i$ and $\exists_L$.

All **LI** rules correspond to a **LK** rule through the erasure of the premises and the conclusions. Hence, we can extend the Erase function from **LI** rules to **LK** rules, and consequently from **LI** proofs to **LK** proofs.

In the same way, we would like to extend the Interpret function from **LI** proofs to **LJ** proofs. This can done associating each **LI** rule to a partial **LJ** proof deriving the interpretation of its conclusion from the interpretation of its premises. However, such an approach would be heavy: as the disjunction in **LJ** is binary, $\bigvee$ is a based on a nesting of binary disjuntions, and a proposition in $\Gamma \vdash \Delta_1; \Delta_2$ can occur deep in $\Gamma, \neg\Delta_2 \vdash \bigvee \Delta_1$. As Interpret will be part of the constructivization algorithm Construct, we need to find another method to define it as a linear-time algorithm.

For this reason, we will define the interpretation of rules using the property that $\Gamma \vdash \bigvee \Delta$ is constructively provable iff $\Gamma, \Delta \Rightarrow G \vdash G$ is provable for any proposition G.

Definition 12. *We define the function* $\text{Interpret}'(\cdot|G)$ *on annotated sequents as* $\text{Interpret}'(\Gamma \vdash \Delta_1; \Delta_2|G) = (\Gamma, \Delta_1 \Rightarrow G, \neg\Delta_2 \vdash G)$.

We extend Interpret$'$ *from* ***LI*** *rules to partial* ***LJ*** *derivations in the following way:*

From a ***LI*** *rule* $\dfrac{\Gamma^1 \vdash \Delta_1^1; \Delta_2^1 \qquad \cdots \qquad \Gamma^n \vdash \Delta_1^n; \Delta_2^n}{\Gamma \vdash \Delta_1; \Delta_2} R$

and a proposition G, we define a partial ***LJ*** *derivation* $\text{Interpret}'(R|G)$ *as a partial derivation of the form*

$$\frac{\text{Interpret}'(\Gamma^1 \vdash \Delta_1^1; \Delta_2^1|G^1) \qquad \cdots \qquad \text{Interpret}'(\Gamma^n \vdash \Delta_1^n; \Delta_2^n|G^n)}{\dfrac{\vdots}{\text{Interpret}'(\Gamma \vdash \Delta_1; \Delta_2|G)}}$$

The ***LI*** *system is designed to ensure that such definitions rely on simple constructive tautologies. As an illustration, we present here the case of the rule*

$$\frac{\Gamma \vdash A, \Delta_1; \Delta_2 \qquad \Gamma, B \vdash \Delta_1; \Delta_2}{\Gamma, A \Rightarrow B \vdash \Delta_1; \Delta_2} \Rightarrow_L^3$$

From a proposition G, defining $\Sigma = \Gamma, \Delta_1 \Rightarrow G, \neg\Delta_2$, we derive

$$\dfrac{\dfrac{}{\Sigma, A \vdash A}\,axiom^* \qquad \dfrac{\Sigma, B \vdash G}{\Sigma, B, A \vdash G}\,weak_L}{\dfrac{\Sigma, A \Rightarrow B, A \vdash G}{\Sigma, A \Rightarrow B \vdash A \Rightarrow G}\,\Rightarrow_R}\,\Rightarrow_L \qquad \dfrac{\Sigma, A \Rightarrow G \vdash G}{\Sigma, A \Rightarrow B, A \Rightarrow G \vdash G}\,weak_L$$
$$\dfrac{\Sigma, A \Rightarrow B \vdash A \Rightarrow G \qquad \Sigma, A \Rightarrow B, A \Rightarrow G \vdash G}{\Sigma, A \Rightarrow B \vdash G}\,cut$$

where the two open premises correspond to $\textsc{Interpret}'(\Gamma, B \vdash \Delta_1; \Delta_2|G)$ *and* $\textsc{Interpret}'(\Gamma \vdash A, \Delta_1; \Delta_2|G)$ *respectively.*

Remark 5. In this case, we chose $G_1 = G_2 = G$. *Other choices for* G_i *appear in the cases* $\wedge_R^2$, $\Rightarrow_L^1$, $\Rightarrow_R^2$, *and* $\forall_R^2$.

In a second step, we extend $\textsc{Interpret}'(\cdot|G)$ *from* ***LI*** *proofs to* ***LJ*** *proofs recursively. Finally, we extend* $\textsc{Interpret}(\cdot)$ *from* ***LI*** *proofs of sequents of the shape* $\Gamma \vdash (G);$ *to* ***LJ*** *proofs:*

– *for* Π *a* ***LI*** *proof of a sequent* $\Gamma \vdash;,$ *we define* $\textsc{Interpret}(\Pi)$ *as*

$$\dfrac{\dfrac{}{\Gamma, \bot \vdash}\,\bot_L^* \qquad \dfrac{\textsc{Interpret}'(\Pi|\bot)}{\Gamma \vdash \bot}}{\Gamma \vdash}\,cut$$

– *for* Π *a* ***LI*** *proof of a sequent* $\Gamma \vdash G;,$ *we define* $\textsc{Interpret}(\Pi)$ *as*

$$\dfrac{\dfrac{\textsc{Interpret}'(\Pi|G)}{\Gamma, G \Rightarrow G \vdash G} \qquad \dfrac{\dfrac{}{\Gamma, G \vdash G}\,axiom^*}{\Gamma \vdash G \Rightarrow G}\,\Rightarrow_R}{\Gamma \vdash G}\,cut$$

Definition 13. *We define the linear-time partial algorithm* $\textsc{Annotate}(\cdot|\cdot)$ *with, as inputs, a* ***LI*** *sequent* S *and a cut-free* ***LK*** *proof* Π *of* $\textsc{Erase}(S)$ *and, as output, either a* ***LI*** *proof of* S *or a failure. This annotation is done from the root to the leaves: at each step, the first argument* S *prescribe how the current conclusion must be annotated. The definition is recursive on the second argument.*

Describing S *as* $\Gamma \vdash \Delta_1; \Delta_2$ *and* Π *as* $\dfrac{\dfrac{\Pi^1}{\Gamma^1 \vdash \Delta^1} \quad \dots \quad \dfrac{\Pi^n}{\Gamma^n \vdash \Delta^n}}{\Gamma \vdash \Delta_1, \Delta_2}\,R$,

– *If there exists a* ***LI*** *rule*

$$\dfrac{\Gamma^1 \vdash \Delta_1^1; \Delta_2^1 \qquad \dots \qquad \Gamma^n \vdash \Delta_1^n; \Delta_2^n}{\Gamma \vdash \Delta_1; \Delta_2}\,R'$$

such that for all i, $\Delta_1^i, \Delta_2^i = \Delta^i$, *then the output is*

$$\dfrac{\dfrac{\textsc{Annotate}(\Gamma^1 \vdash \Delta_1^1; \Delta_2^1|\Pi^1)}{\Gamma^1 \vdash \Delta_1^1; \Delta_2^1} \qquad \dots \qquad \dfrac{\textsc{Annotate}(\Gamma^n \vdash \Delta_1^n; \Delta_2^n|\Pi^n)}{\Gamma^n \vdash \Delta_1^n; \Delta_2^n}}{\Gamma \vdash \Delta_1; \Delta_2}\,R'$$

– *Else,* ANNOTATE$(\cdot,\cdot)$ *fails.*

Remark 6. The only failing cases appear when the rule R is either $\Rightarrow_R$ or $\forall_R$, and exclusively for sequents $\Gamma \vdash \Delta_1; \Delta_2$ such that $|\Delta_1, \Delta_2| > 1$.

Definition 14. *We define the linear-time constructivization algorithm* CONSTRUCT*, which*

- *takes as input a cut-free* ***LK*** *proof* Π *of a sequent* $\Gamma \vdash (G)$*,*
- *computes the proof* $\Pi' = \dfrac{\text{NORMALIZE}(\Pi)}{\Gamma \vdash (G)}\,weak_R$ *,*
- *outputs* INTERPRET(ANNOTATE$(\Gamma \vdash (G); |\Pi')$) *if it exists and fails otherwise.*

Example 1. We consider the law of excluded middle $A \vee \neg A$ given with the following **LK** proof: $\dfrac{\dfrac{\dfrac{}{A \vdash A}\,axiom}{\vdash A, \neg A}\,\Rightarrow_R}{\vdash A \vee \neg A}\,\vee_R$. This proof is unchanged by NORMALIZE.

The ANNOTATE step fails as follows: $\dfrac{\dfrac{\text{FAILURE}}{\vdash A, \neg A;}}{\vdash A \vee \neg A;}\,\vee^1_R$

Example 2. We consider a variant of the non contradiction of law of excluded middle, $(\neg(A \vee \neg A)) \Rightarrow B$, given with the proof:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A, B}\,axiom^*}{\vdash A, \neg A, B}\,\Rightarrow_R}{\vdash A \vee \neg A, B}\,\vee_R \qquad \dfrac{}{\bot \vdash B}\,\bot^*_L}{\neg(A \vee \neg A) \vdash B}\,\Rightarrow_L}{\vdash (\neg(A \vee \neg A)) \Rightarrow B}\,\Rightarrow_R$$

The result of NORMALIZE is

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A}\,axiom}{\vdash A, \neg A}\,\Rightarrow_R}{\vdash A \vee \neg A}\,\vee_R \qquad \dfrac{}{\bot \vdash}\,\bot_L}{\neg(A \vee \neg A) \vdash}\,\Rightarrow_L}{\neg(A \vee \neg A) \vdash B}\,weak_R}{\vdash (\neg(A \vee \neg A)) \Rightarrow B}\,\Rightarrow_R$$

Then, the result of ANNOTATE is

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash ; A}\,axiom^2}{\vdash ; A, \neg A}\,\Rightarrow^2_R}{\vdash ; A \vee \neg A}\,\vee^2_R \qquad \dfrac{}{\bot \vdash ;}\,\bot_L}{\neg(A \vee \neg A) \vdash ;}\,\Rightarrow^1_L}{\neg(A \vee \neg A) \vdash B;}\,weak_R}{\vdash (\neg(A \vee \neg A)) \Rightarrow B;}\,\Rightarrow^1_R$$

As ANNOTATE is the only step which may fail, CONSTRUCT succeeds on this example. We see on the example that the application of NORMALIZE was crucial for ANNOTATE to succeed.

Theorem 4. CONSTRUCT *is complete on F, F_{Ku}, and F_{Ma}: for any proof Π of a sequent S in one of these fragments,* CONSTRUCT(Π) *succeeds.*

Proof. We consider F, F_{Ku}, and F_{Ma} separately:

- For F: we consider a cut-free **LK** proof Π of a sequent $\Gamma \vdash (G) \in F$.

  By Theorem 3, $\Pi' = \dfrac{\text{NORMALIZE}(\Pi)}{\Gamma \vdash (G)}\, weak_R$ is interpretable in **LJ**.

  As a consequence, the only multi-succedent sequents in Π' are conclusions of weakenings. As all failing cases (c.f. Remark 6) involve sequents $\Gamma \vdash \Delta_1; \Delta_2$ such that $|\Delta_1, \Delta_2| > 1$ which are conclusions of $\Rightarrow_R$ or $\forall_R$ rules, ANNOTATE succeeds. Hence, CONSTRUCT succeeds.

- For F_{Ku}: the result follows directly from a stronger assertion: for any cut-free **LK** proof Π of a sequent $\Gamma \vdash \Delta$ containing no $\forall_R$ rule, ANNOTATE($\Gamma \vdash ; \Delta | \Pi$) succeeds. This assertion is proved by induction on such sequents and proofs, noticing that all induction hypotheses refer to sequents of the shape $\Gamma' \vdash ; \Delta'$.

- For F_{Ma}: we consider a cut-free **LK** proof Π of a sequent in F_{Ma}. As mentioned in Remark 6 the only failing cases involve the $\Rightarrow_R$ or $\forall_R$ rules, which don't occur in a proof of a sequent in F_{Ma}. Hence, CONSTRUCT succeeds.

7 Experimental results

In order to measure the success of CONSTRUCT in practice, experiments were made on the basis of **TPTP** [13] first-order problems. The classical theorem prover **Zenon** [10] was used to prove such problems. **Zenon** builds cut-free **LK** proofs internally. It was instrumented to use these internal proofs as inputs for an implementation of WEAK CONSTRUCT and CONSTRUCT. The **LJ** proofs obtained as outputs were expressed and checked in the constructive logical framework **Dedukti** [9].

A set of 724 **TPTP** problems was selected for the experimentations, corresponding to all **TPTP** problems in the category **FOF** which could be proved in less than 1 second using the uninstrumented version of **Zenon**. The results are the following:

- WEAK CONSTRUCT led to constructive proofs in 51% of tested cases.
- CONSTRUCT led to constructive proofs in 85% of tested cases (including all WEAK CONSTRUCT successes).

All constructive proofs generated were successfully checked using **Dedukti**. Among all cases where CONSTRUCT failed, 35% are proved to be unvalid constructively using the constructive theorem prover **ileanCoP** [12].

References

1. Valery Glivenko. Sur quelques points de la logique de m. brouwer. *Bulletins de la classe des sciences*, 15(5):183–188, 1929.
2. Harvey Friedman. Classically and intuitionistically provably recursive functions. In *Higher set theory*, pages 21–27. Springer, 1978.
3. Sigekatu Kuroda et al. Intuitionistische untersuchungen der formalistischen logik. *Nagoya Mathematical Journal*, 2:35–47, 1951.
4. Mélanie Boudard and Olivier Hermant. Polarizing double-negation translations. In Ken McMillan, Aart Middeldorp, and Andreï Voronkov, editors, *LPAR*, volume 8312 of *LNCS ARCoSS*, pages 182–197. Springer, 2013.
5. Shôji Maehara et al. Eine darstellung der intuitionistischen logik in der klassischen. *Nagoya mathematical journal*, 7:45–64, 1954.
6. Gerhard Gentzen. Über das verhältnis zwischen intuitionistischer und klassischer arithmetik. *Archive for Mathematical Logic*, 16(3):119–132, 1974.
7. Kurt Gödel. Zur intuitionistischen arithmetik und zahlentheorie. *Ergebnisse eines mathematischen Kolloquiums*, 4(1933):34–38, 1933.
8. Andrei Nikolaevich Kolmogorov. On the principle of excluded middle. *Mat. Sb*, 32(646-667):24, 1925.
9. M. Boespflug, Q. Carbonneaux, and O. Hermant. The $\lambda\Pi$-calculus modulo as a universal proof language. In David Pichardie and Tjark Weber, editors, *PxTP*, 2012.
10. R. Bonichon, D. Delahaye, and D. Doligez. Zenon: An extensible automated theorem prover producing checkable proofs. In *Logic for Programming, Artificial Intelligence, and Reasoning, 14th International Conference, LPAR 2007, Yerevan, Armenia, October 15-19, 2007, Proceedings*, pages 151–165, 2007.
11. Albert Grigorevich Dragalin and Elliott Mendelson. Mathematical intuitionism. introduction to proof theory. 1990.
12. Jens Otten. leancop 2.0and ileancop 1.2: High performance lean theorem proving in classical and intuitionistic logic (system descriptions). In *Automated Reasoning, 4th International Joint Conference, IJCAR 2008, Sydney, Australia, August 12-15, 2008, Proceedings*, pages 283–291, 2008.
13. G. Sutcliffe. The TPTP Problem Library and Associated Infrastructure: The FOF and CNF Parts, v3.5.0. *Journal of Automated Reasoning*, 43(4):337–362, 2009.